### i Eksamensinformasjon

Eksamensinformasjon – digital skoleeksamen

Fakultet: Teknologi, kunst og design

Utdanning: Teknologiske fag

Emnenavn: Algoritmer og Datastrukturer

Emnekode: (ORD) DATS2300 / ITPE2300

\_\_\_\_\_

Dato: 28.02.2018

Tid: 09.00 - 12.00

Antall oppgaver: 6

Tillatte hjelpemidler: Ingen

#### Merknad:

Råd og tips: Bruk ikke for lang tid på et punkt. Gå isteden videre til neste punkt og eventuelt tilbake hvis du får god tid. Resultatet fra et punkt som du ikke har løst, kan brukes senere i en oppgave som om det var løst. Prøv alle punktene. De 6 deloppgavene teller likt, Hvis du skulle trenge en hjelpestruktur (liste, stakk, kø o.l.) fra java.util eller fra kompendiet, kan du fritt bruke den uten å måtte kode den selv. Men den må brukes på en korrekt måte. Men du bør si fra om dette i en kommentar. Hvis du har idéer om hvordan ting skal løses, men likevel ikke klarer å få det til, kan du demonstrere idéene dine med ord, tegninger o.l.

# <sup>1</sup> Oppgave 1

Metoden med navn ukjent er satt opp i vedlegget. Hva blir utskriften fra følgende kodebit? Gi en forklaring!

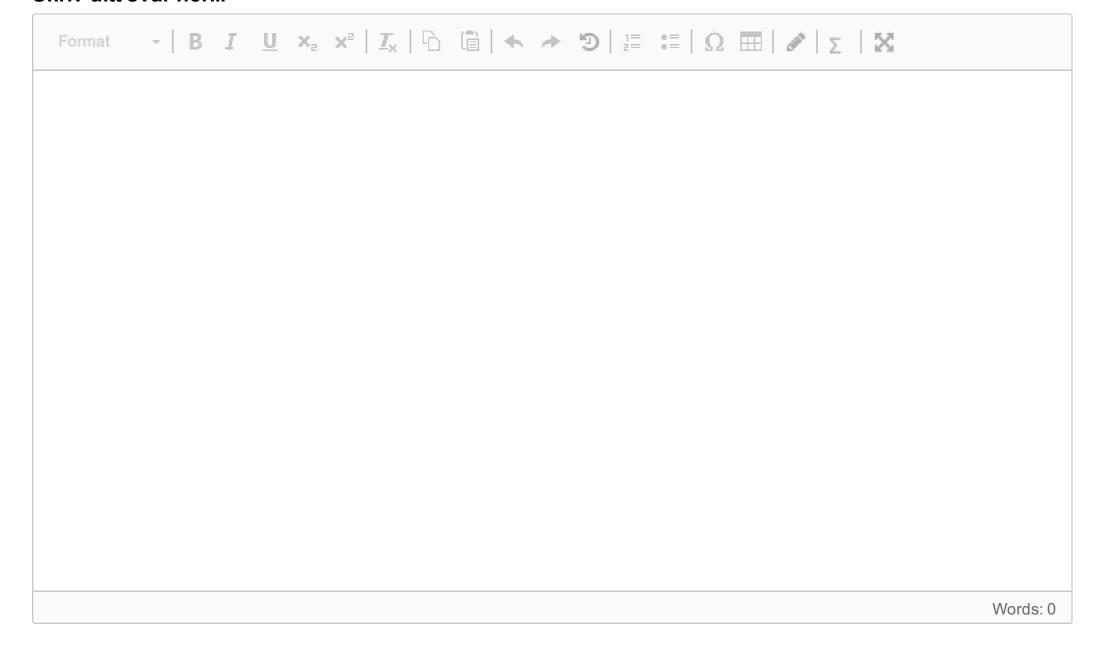
```
int[] a = {1,3,5,7,9};
int[] b = {2,3,5,6,8};
int[] c = new int[a.length + b.length];

int k = ukjent(a,b,c);

for (int i = 0; i < k; i++) System.out.print(c[i] + " ");

Vedlegg:
public static int ukjent(int[] a, int[] b, int[] c)
{
  int i = 0, j = 0, k = 0;
  while ( i < a.length && j < b.length)
  {
    if (a[i] < b[j]) c[k++] = a[i++];
    else if (a[i] == b[j]) { i++; j++; }
    else c[k++] = b[j++];
}
  while (i < a.length) c[k++] = a[i++];
  while (j < b.length) c[k++] = b[j++];
  return k;
}</pre>
```

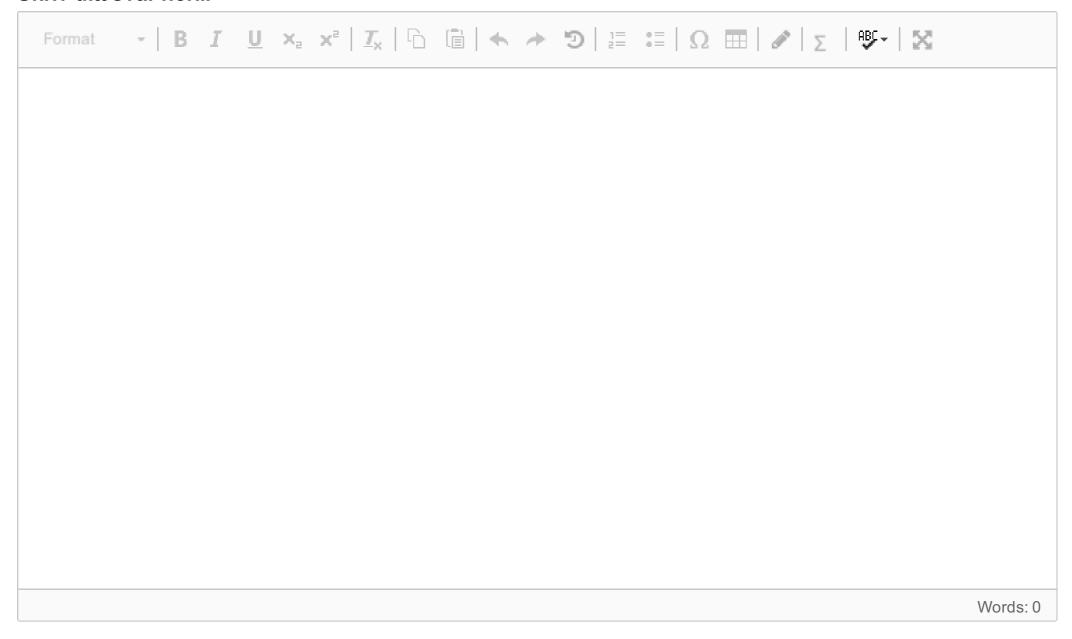
#### Skriv ditt svar her...



#### <sup>2</sup> Oppgave 2

Vi skal komprimere en sekvens med tegn ved hjelp av Huffman-teknikken. Sekvensen inneholder kun tegnene A, B, C, D, E, F, G og H med frekvenser på henholdsvis 49, 16, 2, 29, 115, 17, 30 og 10. Tegn det Huffman-treet dette gir. Sett så opp for hvert av de 8 tegnene den bitkoden som treet bestemmer. Da en liten del av sekvensen ble komprimert ved hjelp av disse bitkodene, ble resultatet: 1101011011010010011011011111. Hvilken delsekvens var det?

Skriv ditt svar her...



Maks poeng: 10

## <sup>3</sup> Oppgave 3

Lag konstruktøren public Mengde(int[] b, int n) (se vedlegget). Tabellen a skal ha lengde n og skal få som innhold de n første verdiene fra parametertabellen b. Hvis de n første verdiene i b ikke er sortert stigende eller inneholder like verdier, skal det kastes en IllegalArgumentException med en tekst. Lag så metoden toString (se vedlegget). Den skal returnere en tegnstreng som inneholder verdiene i a innrammet av hakeparenteser med komma og blank mellom hver verdi. Flg. kodebit viser hvordan dette skal virke:

```
int[] b = {1,2,5,9,11,13,0,0,0,0};
Mengde B = new Mengde(b, 6);  // de 6 første verdiene i b
System.out.println(B);  // et implisitt kall på toString

// Utskrift: [1, 2, 5, 9, 11, 13]

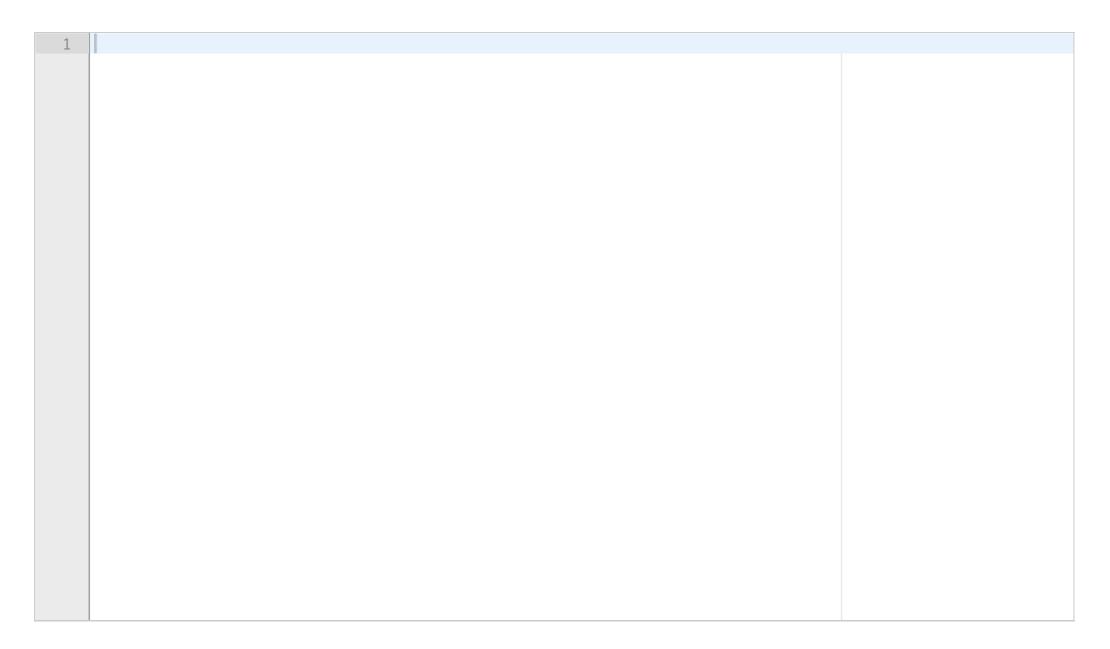
Vedlegg:
public class Mengde
{
    private int[] a;

    public Mengde() // konstruktør
{
        a = new int[0];
    }

public Mengde(int[] b, int n) // konstruktør
{
        // kode mangler - skal lages
```

```
Algoritmer og Datastrukturer DATS2300 / ITPE2300 V2018
}

public String toString()
{
    // kode mangler - skal lages
}
}// class Mengde
Skriv ditt svar her...
```



## <sup>4</sup> Oppgave 4

I denne oppgaven skal du tegne binære søketrær.

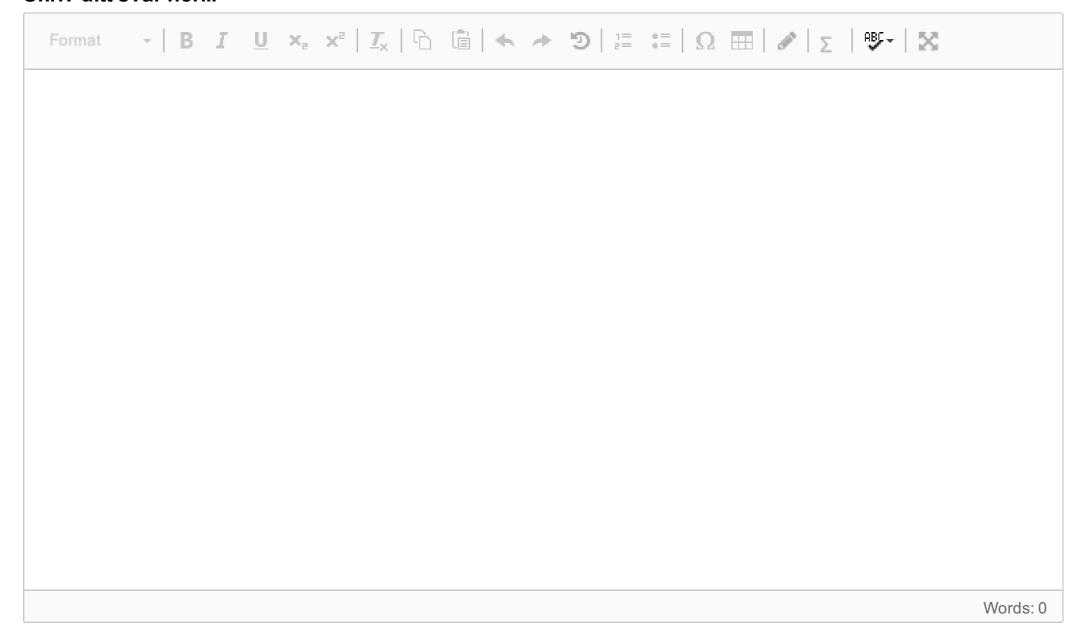
Gitt tallene 16, 4, 8, 12, 7, 11, 15, 6, 7, 2, 3, 5. Legg dem inn, i den gitte rekkefølgen, i et på forhånd tomt *binært søketre*.

a) Tegn treet! Hvilken høyde har treet?

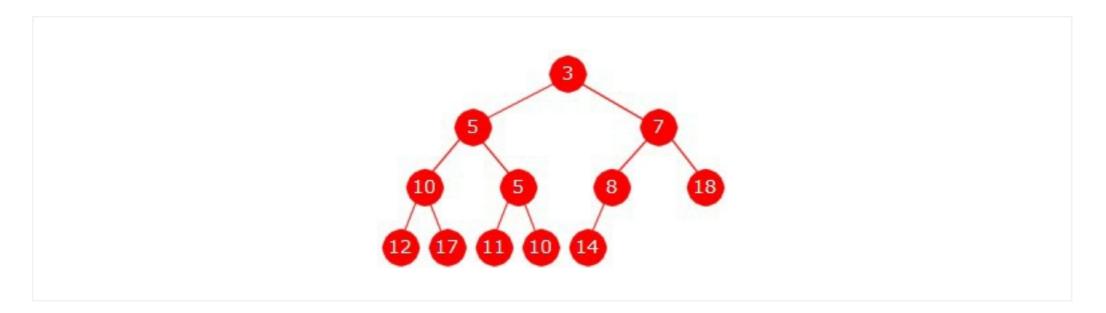
Du skal nå legge inn de samme tallene i et på forhånd tomt 2-3-4-tre.

- b) Tegn treet etter at du har lagt inn 16, 4, 8. Hvilken høyde har dette treet?
- c) Tegn treet etter at du i tillegg har lagt inn 12, 7, 11. Hvilken høyde har dette treet?
- d) Tegn det ferdige treet (etter at alle tallene er lagt inn). Hvilken høyde har dette treet?

Skriv ditt svar her...



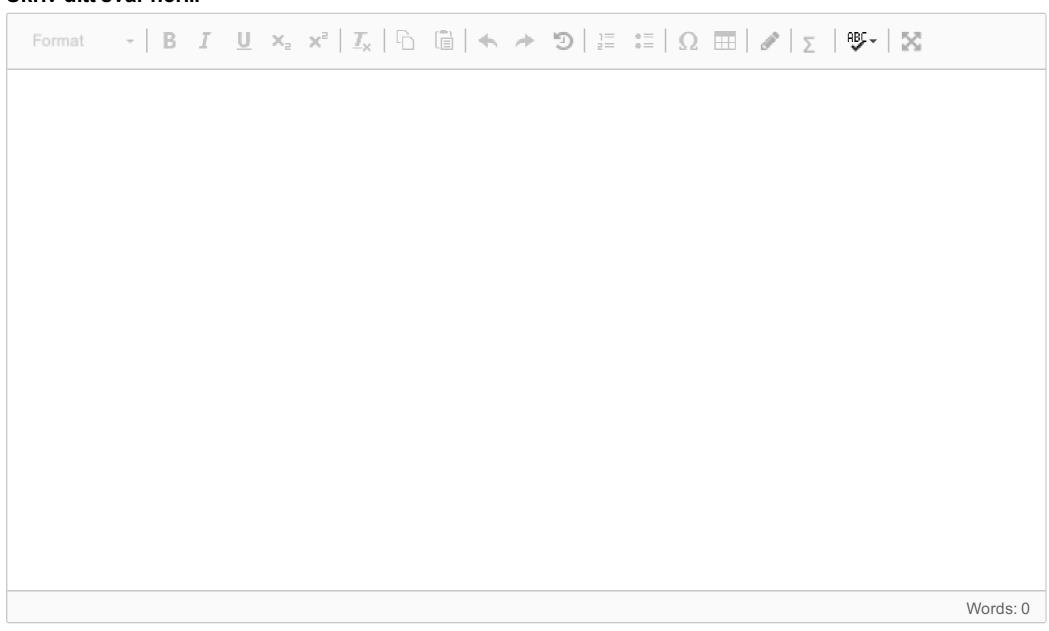
# <sup>5</sup> Oppgave 5



I denne oppgaven skal du jobbe med binær heap som datastruktur. Figuren i vedlagt bilde viser en binær heap (også kalt maksimumstre).

- a) Forklar hva som er spesielt med en binær minimumsheap: hvilke egenskaper må til for at det er en binær minimumsheap?
- b) Hva kan en binær minimumsheap brukes til?
- c) Gitt heap datastrukturen vedlagt bilde, legg til verdiene 6, 10, og 12. Tegn heapen for hvert tall du legger til.
- d) Forklar hvordan en tabell («array») kan brukes for å lagre en minimumsheap i minnet.
- e) Tegn tabellen som tilsvarer heap datastrukturen i figuren.

#### Skriv ditt svar her...



#### <sup>6</sup> Oppgave 6

Klassen <u>LenketHashTabell</u> bruker «lukket adressering med separat lenking». Den inneholder en *tabell* med *nodereferanser* der alle i utgangspunktet er null. Et *objekt* legges inn på objektets *tabellindeks* (objektets hashverdi modulo tabellengden). Dvs. en *node* (med objektet) legges først i den (eventuelt tomme) lenkede nodelisten som hører til tabellindeksen.

En samling navn (tegnstrenger) skal legges inn. Det er en jobb å regne ut hashverdier for hånd. Dette er derfor allerede gjort for noen lengder/dimensjoner:

navn	Espen	Во	Ali	Petter	Karl	Siri	Muhammad	Mari	August	Åse
hashverdi	88008	2357	65964	89125	69562	257197	7934	23763	65085	1983

Setningen LenketHashTabell<String> hash = **new** LenketHashTabell<>(**n**); oppretter en instans av klassen der den interne tabellen får dimensjon (lengde) lik **n**. Legg inn én og én verdi i den gitte rekkefølgen (dvs. Espen, Bo, Ali, osv). En node skal ha både *verdi* og *hashverdi*, men på en tegning holder det med *verdi*.

For enkelthets skyld velger vi å lage en hashtabell av lengde 10. Utfør følgende oppgaver med den lenkede hashtabellen:

- a) Regn ut indeksen for hver hashverdi i hashtabellen.
- b) Lag en tegning av datastrukturen når de fem første navnene er lagt inn
- c) Lag så en tegning som viser når alle navnene er lagt inn. Du trenger ikke å ta hensyn til tetthet (load factor)
- d) Hva betyr begrepet tetthet (load factor) i denne sammenheng, og hva må du gjøre om du skal ta hensyn til dette?
- e) Den lenkede hashtabellen vi har brukt har lengde 10: Er dette en god lengde? Hvorfor / hvorfor ikke? **Skriv ditt svar her...**

